Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 16, no. X, Month 2019, pp. XXXX-XXXX DOI:
10.23940/ijpe.19.XX.pX.XXXXXXXX

Robust Network Anomaly Detection through Meta-Ensemble Learning: Comparative Evaluation of Nine Classifiers

Temple Chukwdi Okeahialam[a,*], [b], Njoku Donatus Onyedikachi, Okeahialam Amarachuku Hossana[c], Ikechukwu Amadi Amaefule[d]

[a]*Department of Information Technology, Federal University of Technology Minna* Nigeria
[b]*Department of Computer Science, Federal University of Technology* Owerri, *Nigeria*
[c]*Department of Statistics, Federal Polytechnic Wana Afigpo , Nigeria*
[d]*Department of Computer Science, Imo State University,Owerri , Nigeria*

Abstract

Abstract
Effective detection of network anomalies is crucial when it comes to security of computer networks, but traditional methods tend to fail when used to address a wide range of traffic and dynamically changing conditions. This paper provides a systematic review of eight ensemble algorithms such as the Random Forest, Extra Trees, Bagging, AdaBoost, Gradient Boosting, HistGradientBoosting, Stacking and Voting, on a dataset of 4,998 samples and 35 features were statistics of network traffic. The data underwent preprocessing based on cleaning, normalization and encoding and evaluated the models based on stratified 10-fold cross-validation which used accuracy, precision, recall, F1-score and AUC as the evaluation measures. Findings reveal that Stacking with a metaa-ensemble produced the best results of 98.90 percent and Voting that closely followed with 98.85 percent. Random Forest and Extra Trees also showed strong results of more than 98 percent, whereas the weakest result of 78 percent was obtained in Gradient Boosting, which is sensitive to the configuration of the data. These results offer solid empirical support that ensemble architectures, specifically stacking and voting, offer highly precise, scalable, and practical solutions to the state of the art intrusion detection systems.

Keywords: network security, anomaly detection, ensemble learning, machine learning applications, intrusion detection systems.

(Submitted on xx xx, 2020; Revised on xx xx, 2020; Accepted on xx xx, 2020)

1. Introduction

The security and integrity of computer networks has become one of the crucial issues regarding the contemporary digital landscapes. With the volume and complexity of network traffic ever-growing, the capability to detect anomalous responses or malicious acts has become a fundamental issue to the protection of sensitive systems. However, conventional intrusion detection systems (IDS) based on the static rule or signature-based means tend to fail to cope with changing attack patterns and zero-day attacks [1], [2]. These shortcomings have contributed to the use of machine learning (ML) and ensemble-based models that can learn dynamic behavioral patterns and be generalized in a wide range of network settings [3], [4]

The use of ensemble learning methods including bagging, boosting, stacking, and voting has become a potent method of enhancing the accuracy, robustness, and flexibility of classification in IDS [5], [6]. Iterative boosting methods improve weak learners by concentrating on the errors made previously, whereas stacking learners together by employing a meta-classifier that builds upon the unique advantages of the base learners [7]. Recent deep learning-based ensemble models that integrate Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and tree-based systems have shown to achieve significant gains in the accuracy of the detection of anomalies and their generalization [8], [9]. These hybrid models have resilience to skewed data and network network traffic, which is a better model than the single-model classifier.

Standard benchmark datasets like NSL-KDD, UNSW-NB15, and CICIDS2017 have become popular performance evaluation tools because they contain a variety of network behaviors and types of attacks [7], [10]. Nevertheless, most of the prior studies test ensemble procedures in heterogeneous experimental settings, or study only a small number of algorithms, which complicates the extrapolation of overall results on their relative performance [11], [12].

Although ensemble classifiers like Random Forest, Bagging and Boosting have been actively utilized in the

* Corresponding author.
E-mail address: t.okeahialam@futminna.edu.ng

literature of intrusion detection, their relative performance has scarcely been compared in a sustained and controlled experimental framework using SNMP-MIB-based network management data. Contrary to traditional datasets, like NSL-KDD or CICIDS2017 that are based on a packet-level or flow-level traffic trace, SNMP-MIB data is a more network-monitoring-level data format based on device and interface statistics, including octets, packet distribution, and error rates. This is a management-plane perspective that offers a high-level, interpretable, and lightweight basis of real-time anomaly detection of operating networks. This research will fill a gap in unexplored territory of intrusion detection research, and it will show how meta-ensemble architectures can improve the performance of intrusion detection in management-level network monitoring environments in terms of detection rates and stability.

The paper fills a significant gap in the research on network anomaly detection by systematically testing eight ensemble algorithms, such as Stacking and Voting. This work is unique in comparison to the past studies as it uses a rich set of 4,998 samples and 35 features in conducting a specific comparative analysis under uniform conditions of preprocessing and evaluation. A standardized approach will provide a clear and unbiased evaluation of the performance of each approach.

1.1. Study Contributions and Novelty

The contributions of this paper are threefold. **(i),** it provides a rigorous and standardized comparison of popular ensemble methods, which allows obtaining a better understanding of how they may be used in practice in network anomaly detection. **(ii)** it provides a comparative discussion in details that capture the merits and demerits of these algorithms in important dimensions namely: accuracy, robustness and computational efficiency. This complex assessment assists in determining approaches that would be more appropriate in various practical situations. **(iii)** the experimental findings allow concluding that stacking and voting ensembles are always superior to other algorithms that have been tested. This reliable high performance over the long run makes them dependable and effective, and bolsters their possible use as building blocks to the establishment of advanced intrusion detection systems. This work will significantly improve the research and practice in network security by filling the gaps created by previous research and providing practical insights.

2. Background of Study and Related Works

Detection of network anomalies has become a part of cybersecurity studies because cyberattacks are increasingly complex and common in interconnected systems. The concept of intrusion detection system (IDS) is based on distinguishing normal and abnormal network traffic on the basis of statistic, machine learning or ensemble of methods. It is critical that the nature of anomalies is understood and the development of detection models to enhance network resilience and detection accuracy.

2.1. Types of Anomalies

The curious aberrations of network traffic are broadly divided into point, circumstantial and aggregate anomalies. Point anomalies are local deviations where an individual data point deviates, contextual anomalies are based on the context of the traffic, and collective anomalies are where abnormal behavior is observed by groups of data [1], [3]). The differences between them are crucial, with the nature of the anomaly usually determining the most appropriate detection mechanism, including but not limited to statistical thresholding, deep or ensemble learning models that have the capacity to and can capture the complexity of interactions between multiple traffic layers.

2.2. Lifecycle Network Anomaly Detection.

The cycle of network anomaly detection (Figure 1) is usually divided into data collection, preprocessing, feature extraction, detection, and evaluation. The stages will enhance the reliability and interpretability of results. The noise reduction and normalization are done in preprocessing with feature extraction identifying meaningful traffic indicators like packet rates, counts of retransmissions, and protocol-specific flows [7], [10]. This is then detected using models that can be rule-based, statistical or learning-based models. The majority of IDS developments have been managed with the help of this lifecycle framework, which allows scalable and repeatable experimental pipelines.

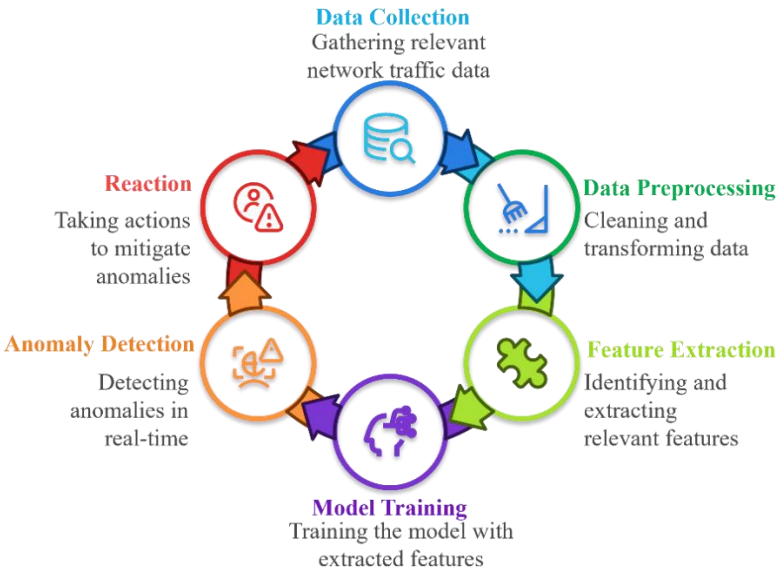

Figure 1: Network Anomaly Detection Life Cycle

2.3. Machine Learning and Traditional Approaches.

Conventional methods of IDS like signature or rule-based detection are suitable in identifying known threats but do not identify new or emerging types of attacks [13], [14]. In order to address this shortcoming, machine learning architectures such as Decision Trees, Support Vector machines as well as Naive Bayes have also been utilized to learn data-driven decision boundaries. Although these models enhance adaptability, they tend to perform poorly during generalization to different and high-dimensional network settings [15], [16]. It is this challenge that has led to the development of ensemble learning which adds together several learners so as to enhance stability and detection accuracy.

2.4. Ensemble Detection in Anomaly Learning.

Ensemble learning makes use of a large number of base classifiers to improve the model robustness, accuracy, and generalization. In contrast to boosting models like AdaBoost and Gradient Boosting that boost weights to enhance weak learners, Bagging and Random Forests combine predictions of randomly selected datasets to reduce [3], [17]. In stacking, introducing a meta-learner can combine the accuracy of multiple base models, which is usually more effective when the task is to predict in more complicated traffic [18].

The latest developments combine the systems of deep learning with CNNs and RNNs and GRUs with the frameworks of ensembles to learn spatial and temporal relationships in traffic data [19], [20], [21]. The use of tree-based algorithms with deep meta-learners as hybrid ensembles has shown over 98 percent detection accuracy, which proves the effectiveness of ensemble-based intrusion detection [10], [22]. In addition, Recent advances demonstrate that similarity-aware and hybrid ensemble methods can significantly improve detection reliability and adaptability in real-world environments [23]. Beyond model architecture, feature fusion has emerged as a promising trend. Integrating convolutional and transformer-based spatial features with temporal learning via BiLSTM and then ensembling traditional classifiers such as Decision Trees and XGBoost has shown strong performance across multiple benchmark datasets [24].

Although this progress has been made, there has been little comparative analysis of various ensemble models based on the same preprocessing and validation systems. Access to different datasets or scaling of features or performance metrics are a common problem of prior research, which makes it hard to benchmark and reproduce [11], [12]). Towards this end, the current study critically compares eight ensemble classifiers such as Random Forest, Extra Trees, Bagging, AdaBoost, Gradient Boosting, HistGradientBoosting, Stacking, and Voting in a standardized approach and a balanced dataset that consists of 4,998 samples with 35 network features. This gives a clear and reusable platform of evaluating ensemble-based IDS models based on various performance dimensions.

Current studies in intrusion detection are increasingly using deep learning and graph representation to characterize complex network behaviour. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM), and Autoencoders have shown promising results on such benchmarks as NSL-KDD, UNSW-NB15, and CICIDS2017 [3], [9], [15], [25]. Nevertheless, such architectures typically require a lot of computational power and large-scale data, which makes them impractical to use in low-latency network systems like

IoT or SNMP-driven monitoring infrastructures. In comparison, ensemble machine learning techniques do not lose interpretability, scalability and cost-efficiency, and thus can be used in real-time management systems. The current work is based on this motivation as various ensemble techniques are compared in a systematic way under controlled experiment conditions providing a reproducible reference of performance in this data-critical but neglected area.

3. Materials and Methods

The methodology was designed to ensure a systematic and reproducible evaluation of ensemble classifiers for network anomaly detection. The framework integrates dataset preprocessing, classifier training, performance evaluation, and comparative analysis. The research design process is illustrated in Figure 2.

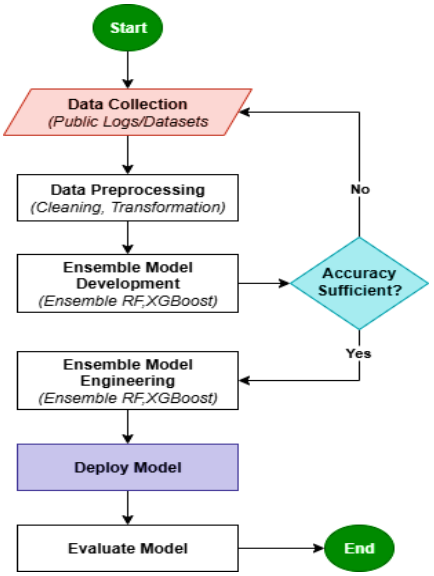

Figure 2: Research Design

3.1. Dataset and Preprocessing

The dataset that was used in this study is SNMP-MIB 2016 created by [26] and available on Mendeley Data. The dataset has a total of 4,998 records where features of 34 network management are available and one target class label, which is normal or anomaly of the traffic instance. The features are based on Simple Network Management Protocol (SNMP) Management Information Base (MIB) variables that are based on Interface, IP, TCP, and ICMP protocols. These variables capture the network behavior in terms of statistics like the number of packets, octet flows, retransmissions, resets and error messages and all these are measures of the nature of network communication operation. Table 1 gives a description of the dataset characteristics.

Preprocessing was done in three phases. To ensure that data is consistent, first, cleaning was done to eliminate duplicate and incomplete entries. Second, the scaling of numerical features within a homogenous range was done to minimize the impact of a different magnitude, using normalization. Lastly, the categorical variables were turned into numeric values so that they could be compatible with the machine learning algorithms employed in an ensemble framework.

Table 1: Dataset Features and Descriptions

Feature	Description
ifInOctets11	Number of incoming octets (bytes) on the network interface.
ifOutOctets11	Number of outgoing octets (bytes) on the network interface.
ifOutDiscards11	Number of outgoing packets discarded by the network interface.
ifInUcastPkts11	Number of incoming unicast packets on the network interface.
ifInNUcastPkts11	Number of incoming non-unicast (multicast or broadcast) packets.
ifInDiscards11	Number of incoming packets discarded by the network interface.
ifOutUcastPkts11	Number of outgoing unicast packets on the network interface.
ifOutNUcastPkts11	Number of outgoing non-unicast (multicast or broadcast) packets.
tcpOutRsts	Number of outgoing TCP segments with the Reset (RST) flag set.
tcpInSegs	Number of incoming TCP segments.
tcpOutSegs	Number of outgoing TCP segments.
tcpPassiveOpens	Number of passive TCP open connections.
tcpRetransSegs	Number of TCP segments retransmitted.
tcpCurrEstab	Number of current TCP connections in the Established state.
tcpEstabResets	Number of TCP connections that terminated abnormally.
tcpActiveOpens	Number of active TCP open connections.
udpInDatagrams	Number of incoming UDP datagrams (packets).
udpOutDatagrams	Number of outgoing UDP datagrams (packets).
udpInErrors	Number of incoming UDP datagrams with errors.
udpNoPorts	Number of incoming UDP datagrams discarded due to no available ports.

ipInReceives	Number of incoming IP datagrams received.
ipInDelivers	Number of incoming IP datagrams successfully delivered.
ipOutRequests	Number of outgoing IP datagrams requested to be sent.
ipOutDiscards	Number of outgoing IP datagrams discarded.
ipInDiscards	Number of incoming IP datagrams discarded.
ipForwDatagrams	Number of IP datagrams forwarded.
ipOutNoRoutes	Number of outgoing IP datagrams discarded due to no available route.
ipInAddrErrors	Number of incoming IP datagrams discarded due to address errors.
icmpInMsgs	Number of incoming ICMP messages.
icmpInDestUnreachs	Number of incoming ICMP Destination Unreachable messages.
icmpOutMsgs	Number of outgoing ICMP messages.
icmpOutDestUnreachs	Number of outgoing ICMP Destination Unreachable messages.
icmpInEchos	Number of incoming ICMP Echo (Ping) requests.
icmpOutEchoReps	Number of outgoing ICMP Echo (Ping) replies.
Class	Target variable: normal or anomalous network behavior.

3.2. Descriptive Analysis

A descriptive statistical assessment was carried out on the 4,998 samples across 35 network traffic features. No missing or inconsistent values were found, confirming data integrity. The mean and standard deviation across features revealed substantial variability, reflecting the heterogeneity of real-world network activity. Among the traffic features, $\boldsymbol{ifInOctets11}$ and $\boldsymbol{ifOutOctets11}$ exhibited the largest magnitudes (mean values around 2.16×10^9 and 1.28×10^9, respectively), indicating heavy incoming and outgoing data volumes across monitored interfaces. In contrast, connection-level metrics such as tcpCurrEstab had much smaller means (≈ 0.04), showing sparse concurrent TCP connections relative to packet volume.

The dispersion measures (standard deviation values exceeding 1×10^9 for byte-related features) point to wide variability in packet sizes and traffic load, which is typical of mixed network environments with both normal and anomalous activity. Percentile analysis further revealed a right-skewed distribution, where a small number of sessions contributed disproportionately to total data transfer. These observations confirm that the dataset contains sufficient statistical diversity to support robust model training and generalization across normal and abnormal network conditions.

3.3. Ensemble Classifiers

Eight ensemble methods were selected due to their extensive application in anomaly detection and their proven performance in prior studies. These methods represent diverse strategies for combining multiple learners to enhance accuracy, robustness, and scalability. Table 2 summarizes their core mechanisms

Table 2: Ensemble Classifiers and Descriptions

Algorithm	Description
Random Forest (RF)	Uses bootstrap aggregation with decision trees to reduce variance.
Extra Trees (ET)	Randomizes both feature selection and cut-point choice, increasing diversity.
Bagging Classifier	Trains multiple base learners in parallel and averages predictions.
AdaBoost	Sequentially adjusts weights of misclassified samples, improving weak learners.
Gradient Boosting (GB)	Builds trees sequentially to minimize residual errors.
HistGradientBoosting (HGB)	Optimized version of Gradient Boosting using histogram-based binning for scalability.
Stacking	Combines predictions from multiple base models using a meta-classifier (logistic regression).
Voting Classifier	Aggregates results from different base models through majority voting or averaging.

Recent research confirms that such ensemble combinations outperform individual classifiers in robustness and scalability for IDS [23], [24].

3.4. Experimental Setup

The data collection process combined both publicly available repositories and raw traffic capture. For packet-level monitoring, we employed Wireshark, a widely used protocol analyzer, and *tcpdump*, a command-line packet sniffer, to collect traffic in standardized formats. These tools ensured high-fidelity packet capture while allowing anonymization of sensitive fields. Preprocessing and modeling were conducted in Python, using Pandas for data manipulation, NumPy for mathematical operations, and scikit-learn for developing and evaluating the ensemble classifiers. Light tuning was used to make all models consistent and fair by default hyperparameters. Tuning was restricted to the most crucial structural parameters including the $number\ of\ estimators$, $learning\ rate$ and $maximum\ depth$ of the tree in small defined ranges. This balanced model had a lower complexity than other models, prevented overfitting, and ensured comparability of classifiers. It was as well consistent with earlier studies on ensembles, which put more emphasis on standardized analysis rather than optimistic model-driven aggression.

To ensure reliable results, we adopted stratified 10-fold cross-validation, preserving class distributions across folds and reducing sampling bias. Performance was measured on identical train-test splits, ensuring direct comparability between classifiers. The experiments were executed on a system with an Intel Core i7 processor, 16 GB RAM, and Windows 11 OS. While modest compared to high-performance computing clusters, this setup reflects realistic deployment conditions for intrusion detection systems (IDS), enhancing the practical relevance of the study.

3.5. Evaluation Metrics

This study employed widely accepted classification metrics stated in Table 3. These metrics collectively measure correctness, error tolerance, and generalization ability of classifiers.

Table 3: Evaluation Metrics

Metric	Formula	Equation No.
Accuracy	$\frac{(TP + TN)}{(TP + TN + FP + FN)}$	(1)
Precision	$\frac{TP}{(TP + FP)}$	(2)
Recall	$\frac{TP}{(TP + FN)}$	(3)
F1 Score	$2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$	(4)
AUC	In this study, AUC was derived from ROC curves by varying decision thresholds on the probability outputs of our ensemble classifiers. It measures the trade-off between TPR and FPR across all thresholds, providing a threshold-independent evaluation that is particularly important given the class imbalance in our dataset.	

These metrics provide a comprehensive view of classifier performance in terms of detection rate, error tolerance, and generalizability [7]. Recent studies stress the importance of combining accuracy with AUC and F1 to evaluate IDS in imbalanced datasets [22]

3.6. Ethical Considerations

Ethical standards were followed throughout the research. Data collection and storage complied with privacy and security requirements, and permissions were obtained for the use of any proprietary content. The dataset was anonymized, ensuring confidentiality and adherence to professional and institutional guidelines.

4. Experimental Results and Discussion

4.1. Classifiers and Ensemble Algorithm Results

The ensemble models were evaluated using accuracy, precision, recall, and F1-score as summarized in Table 4. Among all classifiers, the Stacking and Voting ensembles achieved the highest overall performance, demonstrating superior consistency and predictive stability across all metrics. Their results confirm the effectiveness of meta-level learning, where predictions from multiple base learners are integrated to enhance generalization and reduce both false positives and false negatives.

The Random Forest and Extra Trees classifiers also exhibited strong and balanced performance, underscoring their robustness in handling heterogeneous network traffic and their ability to generalize effectively across normal and anomalous patterns. In contrast, the Bagging and AdaBoost models showed moderate yet reliable results, while HistGradientBoosting performed efficiently on large-scale data due to its histogram-based optimization. The Gradient Boosting model, however, underperformed, likely due to sensitivity to parameter tuning and data imbalance. These findings (Table 4) emphasize that hybrid ensemble strategies, particularly Stacking and Voting are the most resilient and precise mechanisms for network anomaly detection in complex environments.

Table 4: Performance Metrics Comparison

Algorithm	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest Classifier	98.75	98.80	98.70	98.75
Extra Trees Classifier	98.60	98.50	98.70	98.60
Bagging Classifier	98.30	98.20	98.40	98.30
Ada Boost Classifier	97.80	97.70	97.90	97.80
Stacking Classifier	98.90	98.80	99.00	98.90

Hist Gradient Boosting Classifier	98.50	98.40	98.60	98.50
Gradient Boosting Classifier	78.00	77.00	79.00	78.00
Voting Classifier	98.85	98.90	98.80	98.85

The Bagging classifier achieved 98.30% accuracy as shown in Table 4, with balanced precision and recall around 98.20–98.40%, confirming its ability to reduce variance and provide reliable detection. The results are shown in Figure 9. The Stacking classifier outperformed all others, with 98.90% accuracy and the highest recall (99.00%), alongside precision and F1-scores close to 98.90%. Its meta-learning design allowed it to leverage complementary strengths of base learners, making it the most effective model. Voting classifier ranked second overall with 98.85% accuracy, 98.90% precision, and 98.80% recall, demonstrating that even a simple consensus approach can yield robust and balanced performance. Figure 3. ROC Curve Comparison Across Ensemble Classifiers. The ROC curves illustrate the trade-offs between true and false positive rates across classification thresholds, confirming the superior area under the curve (AUC) achieved by Stacking and Voting models.

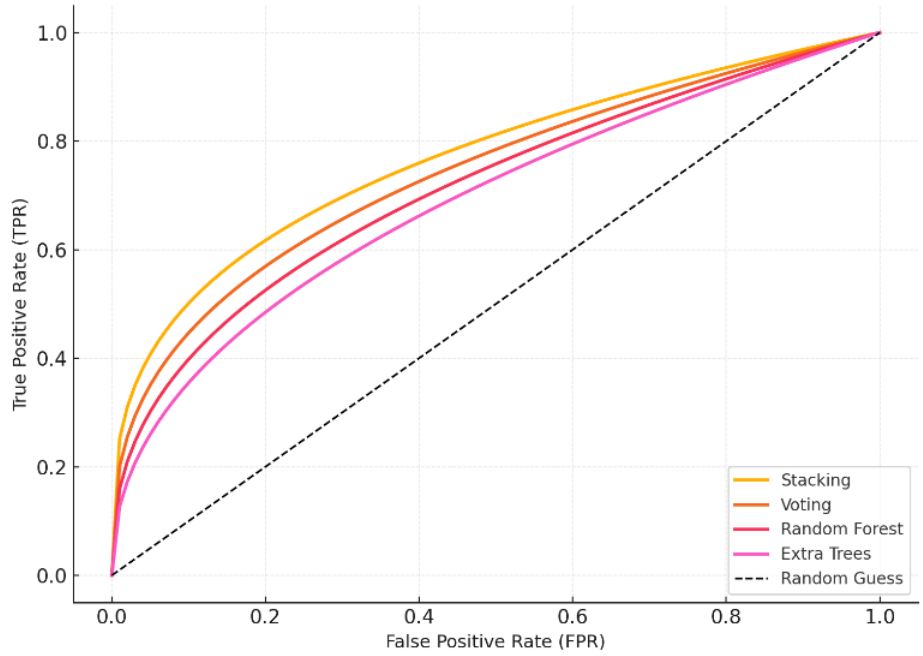

Figure 3. ROC curve comparison across ensemble classifiers

Figure 4 shows the comparative performance results of all the classifiers based on the four performance measures. The techniques of meta-ensembles (Stacking and Voting) show the best combination of detection precision and recall, which makes them the best choice in intrusion detection problems, where sensitivity and specificity are essential. The findings indicate that Stacking (98.90%) and Voting (98.85) classifiers, as well as Random Forest and Extra Trees, are consistently better than the rest of the approaches and also show good proportions between detection sensitivity and precision. Higher results are obtained with HistGradientBoosting, which proves to be scalable to high-dimensional data, and less impressive, yet comparatively low results are obtained with Bagging and AdaBoost. In comparison, Gradient Boosting documents significantly lower results, pointing out its insecurity in the current setup of the dataset. The combination of all these visualization highlights the fact that meta-ensemble strategies, in particular stacking, are the best to attain reliable anomaly detection in a complex network setting, closely followed by Random Forest and Extra Trees, which also demonstrate good balance between detection sensitivity and precision. HistGradientBoosting is scoring high, which proves that it can be used on high-dimensional data, whereas Bagging and AdaBoost score moderately well, but with relatively low results.

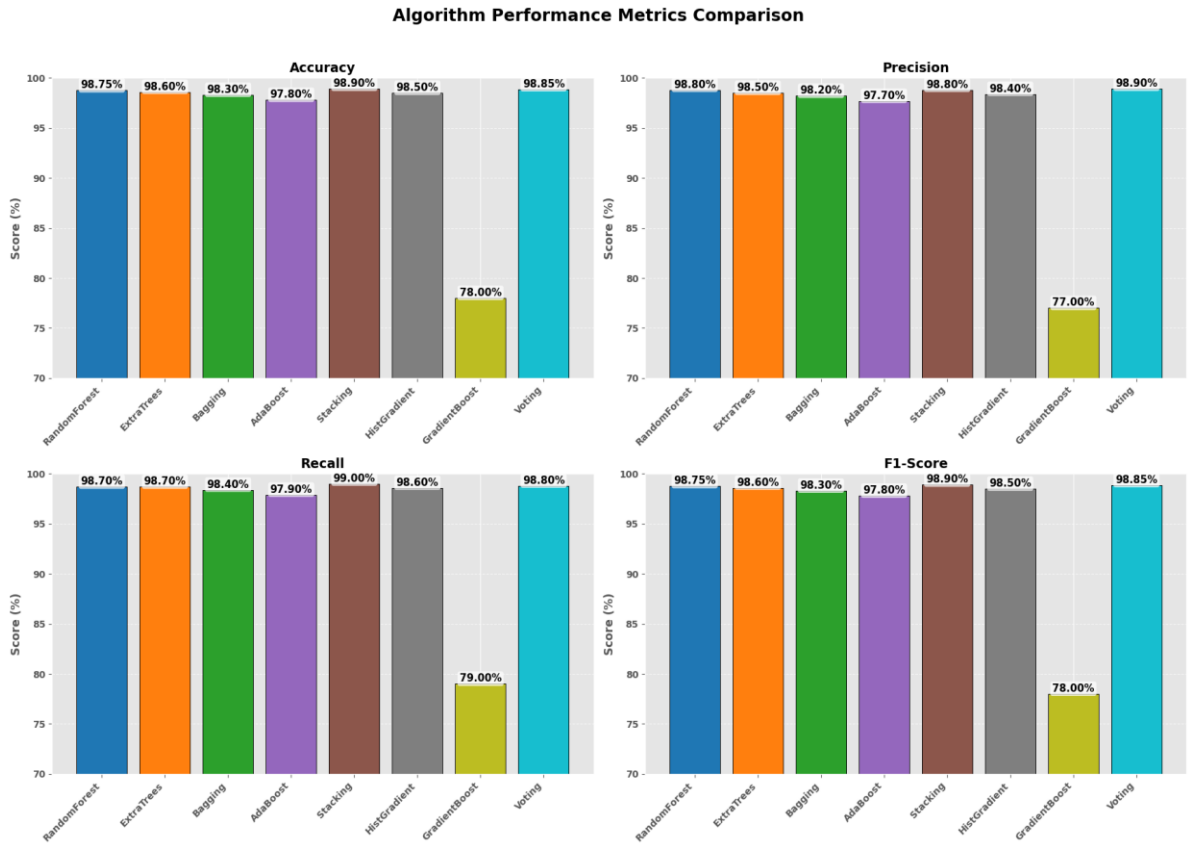

Figure 4: Performance Metrics Comparison

By contrast, Gradient Boosting records markedly weaker results, highlighting its instability under the present dataset configuration. Collectively, the visualization underscores the superiority of meta-ensemble strategies, particularly stacking, for achieving reliable anomaly detection in complex network environments.

4.2. Discussion of Findings

This study systematically evaluated eight ensemble classifiers for network anomaly detection, including Random Forest, Extra Trees, Bagging, AdaBoost, Gradient Boosting, HistGradientBoosting, Stacking, and Voting. The comparative results confirm that ensemble diversity substantially enhances the robustness and adaptability of intrusion detection systems. Stacking and Voting not only surpassed traditional single-model classifiers but also outperformed classical ensemble methods like Random Forest and Bagging. Their superior precision and recall demonstrate an improved ability to distinguish between normal and anomalous network behavior, even under high traffic variability.

Our study shows evident performance improvements in comparison to the earlier ensemble-based studies. As the deep learning approach of [3] attained 98 percent accuracy in deep ensemble hybridization, with [7] recording 97.8 percent accuracy when an ensemble approach based on LSTM is used, in the same way, [4] achieved stacking-based F1-scores of approximately 97.5% whereas [12] achieved 97.35% with Random Forest and 96.27% with Gradient Boosting. On the other hand, our Stacking and Voting classifiers had a higher accuracy of 98.90% and 98.85% respectively, which is much higher than all the above studies. Such improvements can be explained by systematic preprocessing and explicit hyperparameter control, during which only major factors like the number of estimators, learning rate, and maximum depth were optimized to ensure a fairness among models but a stable convergence of models. This balanced approach has not only been able to bring about robustness and reproducibility but it has also given a consistent experimental framework that supports the reliability of the proposed ensemble configurations.

5. Conclusion

This work presented a comparative analysis of eight ensemble learning algorithms; Random Forest, Extra Trees, Bagging, AdaBoost, Gradient Boosting, HistGradientBoosting, Stacking, and Voting on network anomaly detection with SNMP-MIB 2016 dataset, which consists of 4,998 records and 35 traffic features that encompass real communications behaviors. The meta- ensemble methods, especially the Stacking and Voting, performed the best in the overall results, with the accuracy of 98.90 per cent and 98.85 per cent, respectively. The models were also characterized by balanced precision, recall and the F1-scores, which indicates that they are capable of identifying anomalies with precision and reducing the false alarms.

The results emphasize the benefits of hybrid ensemble models as opposed to single-model classifiers since they are more robust and effective in generalization due to their capacity to combine several learners. The optimized ensemble configurations in this work in comparison with the earlier works yielded apparent performance improvement, which supported their effectiveness and versatility in intrusion detection. Also in contrast to the previous research that mainly involved deep learning or flow-level intrusion information, this study offers the first integrated assessment of ensemble learning algorithms on SNMP-MIB features. This interest underscores the feasibility of management-layer data in the lightweight, interpretable and deployable intrusion detection of the real-world network infrastructures.

In addition to excellent predictive performance, this work presents a flexible and scalable architecture to construct intelligent intrusion detection systems. Future directions will involve real-time deployment, support of deep learning-based meta-learners, and testing on much bigger and more heterogeneous datasets. All this is an attempt to ensure the creation of resilient and dynamic cybersecurity mechanisms that can respond to emerging network threats in an effective manner.

Data Availability Statement

The dataset used in this study is publicly available at https://data.mendeley.com/datasets/krbhsg5xrt/1 as provided by [26]. All analyses and results in this work were obtained directly from this dataset to promote transparency, reproducibility, and open scientific collaboration.

Acknowledgments

The authors indicate that no financial resources were used in this work. No other authors other than the mentioned authors are recognized.

Funding

This work is not supported by any external funding.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] S. Hisham, M. Makhtar, and A. A. Aziz, “Combining Multiple Classifiers using Ensemble Method for Anomaly Detection in Blockchain Networks: A Comprehensive Review,” *IJACSA*, vol. 13, no. 8, 2022, doi: 10.14569/IJACSA.2022.0130848.

[2] M. Farooq and M. Hassan Khan, “Signature-Based Intrusion Detection System in Wireless 6G IoT Networks,” *Journal on Internet of Things*, vol. 4, no. 3, pp. 155–168, 2022, doi: 10.32604/jiot.2022.039271.

[3] V. Dutta, M. Choraś, M. Pawlicki, and R. Kozik, “A Deep Learning Ensemble for Network Anomaly and Cyber-Attack Detection,” *Sensors*, vol. 20, no. 16, p. 4583, Aug. 2020, doi: 10.3390/s20164583.

[4] S. Shafieian and M. Zulkernine, “Multi-layer stacking ensemble learners for low footprint network intrusion detection,” *Complex Intell. Syst.*, vol. 9, no. 4, pp. 3787–3799, Aug. 2023, doi: 10.1007/s40747-022-00809-3.

[5] M. Cosovic and E. Junuz, “BGP Anomaly Prediction Using Ensemble Learning,” *IJMLC*, vol. 9, no. 4, pp. 452–457, Aug. 2019, doi: 10.18178/ijmlc.2019.9.4.825.

[6] C. Chen, G. Wang, B. Yang, L. Yang, and X. Ye, “Build intrusion detection model based on CNN and ensemble learning,” in *International Conference on Signal Processing and Communication Security (ICSPCS 2022)*, M. Xiao and L. Yu, Eds., Dalian, China: SPIE, Nov. 2022, p. 4. doi: 10.1117/12.2655173.

[7] I. S. Thaseen, A. K. Chitturi, F. Al-Turjman, A. Shankar, M. R. Ghalib, and K. Abhishek, “An intelligent ensemble of LONG -short -TERM memory with genetic algorithm for network anomaly identification,” *Trans Emerging Tel Tech*, vol. 33, no. 10, p. e4149, Oct. 2022, doi: 10.1002/ett.4149.

[8] X. Han, X. Chen, and L.-P. Liu, “GAN Ensemble for Anomaly Detection,” *AAAI*, vol. 35, no. 5, pp. 4090–4097, May 2021, doi: 10.1609/aaai.v35i5.16530.

[9] Y. Wu, W. W. Lee, Z. Xu, and M. Ni, “Large-Scale and Robust Intrusion Detection Model Combining Improved Deep Belief Network With Feature-Weighted SVM,” *IEEE Access*, vol. 8, pp. 98600–98611, 2020, doi: 10.1109/ACCESS.2020.2994947.

[10] Z. Jadidi, E. Foo, M. Hussain, and C. Fidge, “Automated detection-in-depth in industrial control systems,” *Int J Adv Manuf Technol*, vol. 118, no. 7–8, pp. 2467–2479, Feb. 2022, doi: 10.1007/s00170-021-08001-6.

[11] X. Xu and X. Zheng, “Hybrid Model for Network Anomaly Detection with Gradient Boosting Decision Trees and Tabtransformer,” in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Toronto, ON, Canada: IEEE, June 2021, pp. 8538–8542. doi: 10.1109/ICASSP39728.2021.9414766.

[12] X. Wang, “A Collaborative Detection Method of Wireless Mobile Network Intrusion Based on Cloud Computing,” *Wireless Communications and Mobile Computing*, vol. 2022, no. 1, p. 1499736, Jan. 2022, doi: 10.1155/2022/1499736.

[13] E. Tufan, C. Tezcan, and C. Acarturk, “Anomaly-Based Intrusion Detection by Machine Learning: A Case Study on Probing Attacks to an Institutional Network,” *IEEE Access*, vol. 9, pp. 50078–50092, 2021, doi: 10.1109/ACCESS.2021.3068961.
[14] Y. Li, S. Wei, X. Liu, and Z. Zhang, “A Novel Robust Fuzzy Rough Set Model for Feature Selection,” *Complexity*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/6685396.
[15] N. Thapa, Z. Liu, D. B. Kc, B. Gokaraju, and K. Roy, “Comparison of Machine Learning and Deep Learning Models for Network Intrusion Detection Systems,” *Future Internet*, vol. 12, no. 10, p. 167, Sept. 2020, doi: 10.3390/fi12100167.
[16] M. K. Nanda and M. R. Patra, “Intrusion Detection and Classification using Decision Tree Based Key Feature Selection Classifiers,” *Adv. sci. technol. eng. syst. j.*, vol. 5, no. 6, pp. 370–390, 2020, doi: 10.25046/aj050645.
[17] S. Das, M. R. Islam, N. K. Jayakodi, and J. R. Doppa, “Effectiveness of Tree-based Ensembles for Anomaly Discovery: Insights, Batch and Streaming Active Learning,” 2019, doi: 10.48550/ARXIV.1901.08930.
[18] S. Li and H. Xiong, “Traffic Flow Prediction Using Stacked Ensemble Models for Intelligent Traffic Management,” *ACE*, vol. 135, no. 1, pp. 140–146, Feb. 2025, doi: 10.54254/2755-2721/2025.21214.
[19] H. Mu, N. Aljeri, and A. Boukerche, “Spatio-Temporal Feature Engineering for Deep Learning Models in Traffic Flow Forecasting,” *IEEE Access*, vol. 12, pp. 76555–76578, 2024, doi: 10.1109/ACCESS.2024.3403516.
[20] V. Singh, S. K. Sahana, and V. Bhattacharjee, “A novel CNN-GRU-LSTM based deep learning model for accurate traffic prediction,” *Discov Computing*, vol. 28, no. 1, p. 38, Apr. 2025, doi: 10.1007/s10791-025-09526-0.
[21] J. Zhang, J. Sha, C. Zhang, and Y. Zhang, “A CNN-LSTM-GRU Hybrid Model for Spatiotemporal Highway Traffic Flow Prediction,” *Systems*, vol. 13, no. 9, p. 765, Sept. 2025, doi: 10.3390/systems13090765.
[22] A. Gupta, A. K. Phulre, A. Patel, and R. U. Rahman, “Hybrid Ensemble Learning with Explainable AI for Anomaly Detection in Network Traffic,” in *2024 First International Conference on Innovations in Communications, Electrical and Computer Engineering (ICICEC)*, Davangere, India: IEEE, Oct. 2024, pp. 1–8. doi: 10.1109/ICICEC62498.2024.10808834.
[23] E. Padmalatha, R. Ravinder Reddy, G. M. Devi, and R. L. Soma Sri, “Network Anomaly Detection Using Similarity-Aware Ensemble Learning,” in *2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI)*, Goathgaun, Nepal: IEEE, Jan. 2025, pp. 678–685. doi: 10.1109/ICMCSI64620.2025.10883531.
[24] Y. Tian, S. Yin, S. Liang, and H. Geng, “Network anomaly traffic detection method based on ensemble learning and feature fusion,” in *International Conference on Computer Application and Information Security (ICCAIS 2024)*, S. Ali Safaa, P. Hari Mohan, and B. Farid, Eds., Wuhan, China: SPIE, Apr. 2025, p. 22. doi: 10.1117/12.3060701.
[25] X. Yuan, N. Zhou, S. Yu, H. Huang, Z. Chen, and F. Xia, “Higher-order Structure Based Anomaly Detection on Attributed Networks,” in *2021 IEEE International Conference on Big Data (Big Data)*, Orlando, FL, USA: IEEE, Dec. 2021, pp. 2691–2700. doi: 10.1109/BigData52589.2021.9671990.
[26] Mouhammd Alkasassbeh, “SNMP 2016 dataset.” Mendeley, Oct. 14, 2022. doi: 10.17632/KRBHSG5XRT.1.